



Planet Detection Metrics:

Window and One-Sigma Depth Functions
for Data Release 25

KSCI-19101-002

Christopher J. Burke and Joseph Catanzarite
03 May 2017

**NASA Ames Research Center
Moffett Field, CA 94035**

Prepared by:


Christopher J. Burke, Science OfficeDate 5/3/17

Prepared by:

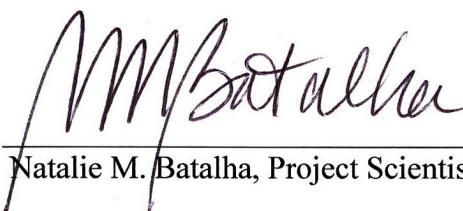

Joseph Catanzarite, Science Operations CenterDate 5/3/17

Approved by:


Michael R. HaasDate 5/3/17

Michael R. Haas, Science Office Director

Approved by:


Natalie M. BatalhaDate 5/3/17

Natalie M. Batalha, Project Scientist

Document Control

Ownership

This document is part of the *Kepler* Project Documentation that is controlled by the *Kepler* Project Office, NASA/Ames Research Center, Moffett Field, California.

Control Level

This document will be controlled under KPO @ Ames Configuration Management system. Changes to this document **shall** be controlled.

Physical Location

The physical location of this document will be in the KPO @ Ames Data Center.

Distribution Requests

To be placed on the distribution list for additional revisions of this document, please address your request to the *Kepler* Science Office:

Michael R. Haas
Kepler Science Office Director
MS 244-30
NASA Ames Research Center
Moffett Field, CA 94035-1000

or

Michael.R.Haas@nasa.gov

DOCUMENT CHANGE LOG

CHANGE DATE	PAGES AFFECTED	CHANGES/NOTES
December 1, 2016	All	Original release
May 3, 2017	11, 14, 15-18	Added warning about NaNs in one-sigma depth functions, updated references, and added Appendix A to define related products.

Table of Contents

1.	Introduction.....	6
2.	Window Function.....	7
3.	One-Sigma Depth Function	9
4.	Window Function Changes and Verification.....	12
5.	References.....	14
	Appendix A: Column Definitions for Related Products	15

1. Introduction

This document describes the window and one-sigma depth functions relevant to the Transiting Planet Search (TPS) algorithm in the *Kepler* pipeline (Jenkins 2002; Jenkins et al. 2017). The window function specifies the fraction of unique orbital ephemeris epochs over which three transits are observable as a function of orbital period. In this context, the epoch and orbital period, together, comprise the ephemeris of an orbiting companion, and ephemerides with the same period are considered equivalent if their epochs differ by an integer multiple of the period. The one-sigma depth function specifies the depth of a signal (in ppm) for a given light curve that results in a one-sigma detection of a transit signature as a function of orbital period when averaged over all unique orbital ephemerides.

These planet detection metrics quantify the ability of TPS to detect a transiting planet signature on a star-by-star basis. They are uniquely applicable to a specific *Kepler* data release, since they are dependent on the details of the light curves searched and the functionality of the TPS algorithm used to perform the search. This document describes the window and one-sigma depth functions relevant to *Kepler* Data Release 25 (DR25), where the data were processed (Thompson et al. 2016) and searched (Twicken et al. 2016) with the SOC 9.3 pipeline. In Section 4, we describe significant differences from those reported in *Kepler* Data Release 24 (Burke & Seader 2016) and document our verification method.

These data products are archived at the NExSci Exoplanet Archive¹ as two separate FITS files per target as described below. They are available for all targets searched for planets by the *Kepler* pipeline (Twicken et al. 2016) and can be downloaded as a set, or for individual targets through the interactive interface.

¹ <http://exoplanetarchive.ipac.caltech.edu/index.html>

2. Window Function

The window function for a transit survey specifies the fraction of unique orbital ephemeris epochs that permits $\geq k$ transits to be observed as a function of period for a fixed-length data set (Gaudi 2000; Burke et al. 2006; Burke & McCullough 2014). In this context, ephemerides with the same period are considered identical if their epochs differ by an integer multiple of the period. The *Kepler* pipeline requires $k \geq 3$ transits for planet detection at each of the fourteen transit durations searched (*i.e.*, 1.5, 2.0, 2.5, 3.0, 3.5, 4.5, 5.0, 6.0, 7.5, 9.0, 10.5, 12.0, 12.5, and 15.0 hrs), so the reported window functions are based on this three-transit criteria. An additional check is performed by TPS when exactly three transits contribute to a putative detection. In the three-transit case only, each transit is inspected. If less than half of the in-transit cadences have full weight (*e.g.*, full weight, $w=1.0$, and full deweighting, $w=0.0$) for any of the three transits that contribute to the detection, then the detection is rejected and does not become a Threshold Crossing Event (TCE). The three-transit weight check was not included in the DR24 calculation of the window function (Burke & Seader 2016). In Section 4, we discuss the impact of including this additional check.

The FITS files are named `kplr<identifier>_<release>_window.fits`, where `<identifier>` corresponds to the target's *Kepler* Input Catalog (KIC) identification number and `<release>` specifies the *Kepler* data release. Each file contains 14 binary table extensions for the 14 pulse durations searched by TPS. The transit duration is specified by the keyword TDUR in the FITS header for each extension. Each binary table extension has two columns corresponding to the orbital period (in days) and the window function (unitless) for the stated transit duration. In order to evaluate the window function at an arbitrary orbital period, it is up to the end user to interpolate between the tabulated orbital periods. However, in practice the resolution in orbital period provided is sufficient for ‘nearest neighbor’ interpolation in most use cases.

Due to the expected increase in transit duration with orbital period for transiting planet signatures, the entire period range is not searched at all transit durations. The shortest transit durations have an upper limit to the orbital periods searched, and the longest transit durations have a lower limit to the orbital periods searched. Thus, the range of orbital period over which valid window functions are computed by TPS varies with duration. Section 2.2 of Tenenbaum et al. (2014) describes details of the orbital period range searched as a function of transit duration. The orbital period range accommodates the expected transit durations assuming circular orbits with extended coverage to allow for moderately eccentric orbits.

Several example window functions are provided in the left-hand panels of Figures 1 – 3. These examples illustrate targets that have varying amounts of observational data available. The window function ranges from 0 to 1. It has a value of one at short periods, where there is a high probability of having at least three transits. The window function then decays to zero at long orbital periods, since there is a decreasing probability that the flux time series is long enough to allow at least three transits to be observed. For nearly all *Kepler* targets, the window function varies insignificantly with transit duration,

so it is typically valid to adopt the results of any small set of durations that spans the full range of periods. This is well illustrated in the left-hand panels of Figures 1 – 3, where all 14 durations are plotted, but only one continuous curve is apparent.

3. One-Sigma Depth Function

The detectability of a transit signal is predominately determined by its signal-to-noise ratio (SNR) (Gaudi 2000). One can estimate the transit SNR as $\Delta \sqrt{N_{\text{tran}}} / \sigma_{\text{cdpp}}$, where Δ is the expected transit depth, N_{tran} is the expected number of transits, and σ_{cdpp} is an averaged noise estimate from the root-mean-square (rms) CDPP (Christiansen et al. 2012). However, this SNR estimate assumes all observations have the same noise properties and that missing or de-weighted data in the flux time series are evenly distributed in time. For the one-sigma depth function, TPS quantifies the transit signal depth that yields a SNR of unity as a function of the orbital period by taking into account the full details of the time varying properties of the noise and accounting for any missing or de-weighted data. When using the one-sigma depth function, Δ_{one} , the transit signal-to-noise ratio is simply $\text{SNR} = \Delta / \Delta_{\text{one}}$.

As for the window function, the one-sigma depth function is archived in FITS format. Each file contains 14 binary table extensions giving results for the 14 transit durations searched by TPS. Each binary table extension has two columns corresponding to the orbital period (in days) and the one-sigma depth (in ppm). The file names have the form `kplr<identifier>_<release>_onesigdepth.fits`, where `<identifier>` and `<release>` have the same definitions as for the window function files. In order to evaluate the one-sigma depth function at an arbitrary orbital period, it is up to the end user to interpolate between the tabulated orbital periods. However, in practice the resolution in orbital period provided is sufficient for ‘nearest neighbor’ interpolation in most use cases.

Several examples of one-sigma depth functions are provided in the right-hand panels of Figures 1 - 3. These examples feature targets with varying amounts of missing data to illustrate the impact of incompleteness. In the large N_{tran} (*i.e.*, short-period) regime, Δ_{one}

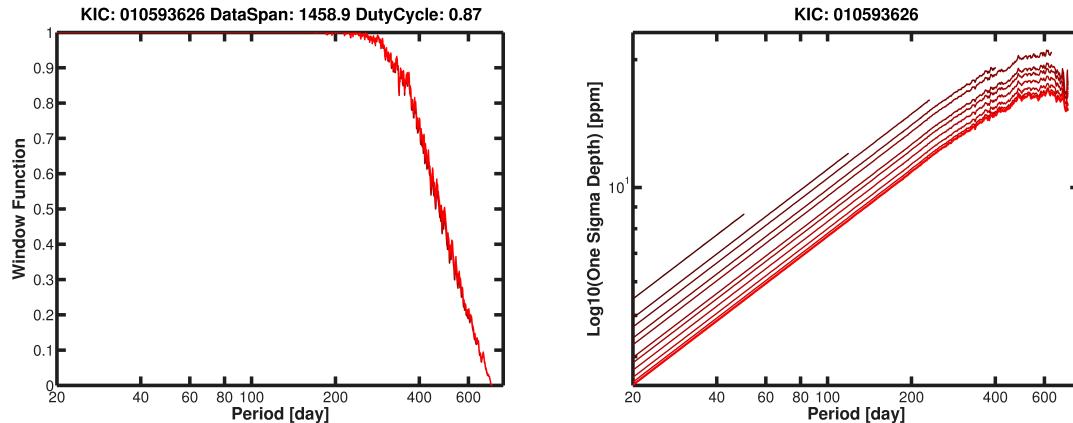


Figure 1. Example planet detection metrics for a target with data available in every quarter (Q1-Q17). This target corresponds to KIC 10593626, the host of *Kepler-22b* (Borucki et al. 2012). **Left:** Window function as a function of orbital period. **Right:** One-sigma depth function as a function of orbital period. The top (bottom) line corresponds to the shortest (longest) transit duration searched for transit signatures.

increases as $\sqrt{P_{\text{orb}}}$, where P_{orb} is the orbital period. For small values of N_{tran} (*i.e.*, long periods), Δ_{one} deviates from this simple dependence on P_{orb} because the noise properties of the full flux time series are not well represented by such sparse sampling.

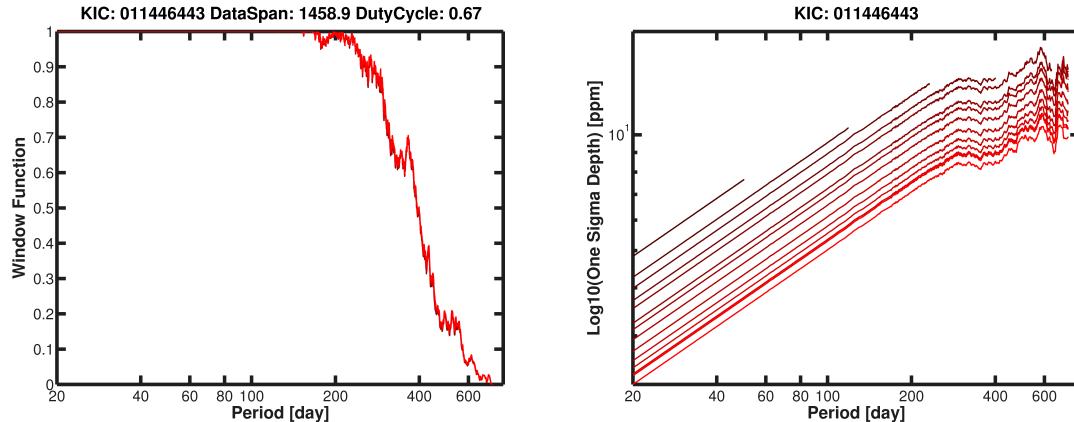


Figure 2. Example planet detection metrics for a target impacted by the loss of Module 3 (*i.e.*, every fourth quarter of data is missing). This target corresponds to KIC 11446443, the host of TrES-2b (O'Donovan et al. 2006). **Left:** Window function as a function of orbital period. **Right:** One-sigma depth function as a function of orbital period. The top (bottom) line corresponds to the (shortest) longest transit duration searched for transit signatures.

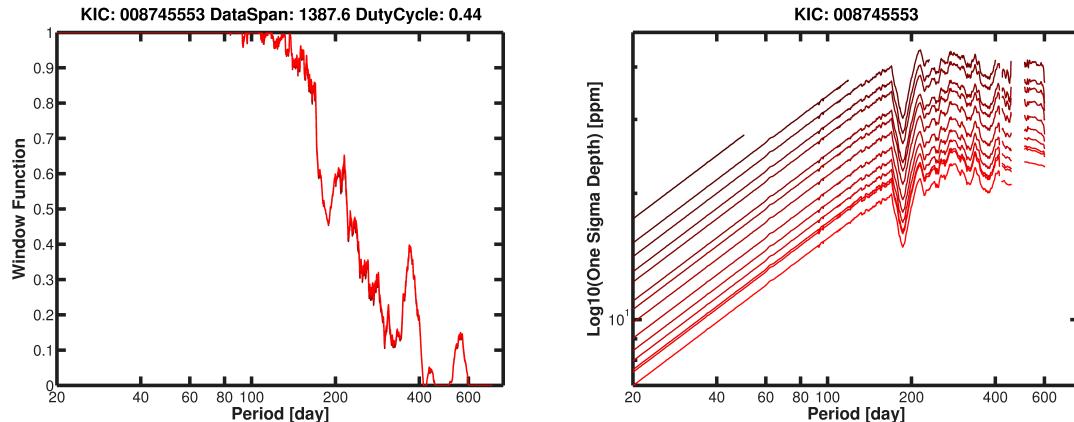


Figure 3. Example planet detection metrics for a target with data missing every other quarter. This target corresponds to KIC 8745553. In general, targets with significant amounts of missing data or patterns in the missing data segments have increased structure in the planet detection metrics as data coherence rapidly changes with orbital period. **Left:** Window function as a function of orbital period. **Right:** One-sigma depth function as a function of orbital period. The top (bottom) line corresponds to the (shortest) longest transit duration searched for transit signatures.

WARNING: Note that the FITS binary tables for the one-sigma depth function contain not-a-number (NAN) entries for orbital periods where there is insufficient data for the minimum three-transit-event detection requirement (*i.e.*, the window function goes to zero). In order to prevent these NAN entries from propagating further, we recommend that end-users replace them with a suitably high number (*e.g.*, 10^6 ppm) immediately after reading in the files such that relative flux decrements of 100% cannot make a significant detection. The target KIC 874553, shown in Figure 3, is impacted by such NAN entries as illustrated by the gap in the one-sigma depth function data over the 400-600 day orbital period range (right panel) where the window function has zero detectability (left panel).

4. Window Function Changes and Verification

In this section, we briefly describe the method used to calculate the window functions and their verification using an alternative Monte-Carlo sampling method. For *Kepler*, the window function requires evaluating, for a given ephemeris (*i.e.*, period and epoch) whether enough data exists to observe three independent transit events and whether the in-transit data passes the three-transit weight check described in Section 2. The results depend on the details of the TPS version employed. Twicken et al. (2016) and Jenkins et al. (2017) describe the SOC 9.3 pipeline employed for DR25.

In principle, the window function could be computed by TPS during the actual transit search within the pipeline. However, in earlier versions, the requisite information was not retained between evaluations at different ephemerides. Therefore, the signal search function in TPS was augmented post facto to accumulate this information across all ephemeris evaluations. With the penalty of a few storage variables, the window function was then calculated outside the formal pipeline with this modified version of TPS.

In parallel with this effort, we developed a Monte-Carlo transit injection and recovery test for TPS known as flux-level transit injection (FLTI). This FLTI uses a highly scaled-down version of TPS that only searches the ephemeris grid points enclosing the injected ephemeris location, rather than the entire period grid of a normal search. In brief, this FLTI code adds realistic limb-darkened transit signals to the flux time series that was previously quarter-stitched and detrended during the full TPS run (see Thompson 2016). The FLTI is based on TPS alone and is completely independent of the other pipeline modules (Jenkins et al. 2017). Hence, this is very different (and many orders of magnitude faster) than the pixel-level transit injection study of Christiansen et al. (2016), which injects transit signals into the calibrated (CAL) pixel-level data and runs the *Kepler* pipeline to perform the photometry (PA), systematic error correction (PDC), full transit search (TPS), and signal evaluation (DV). These shortcuts allow us to test the recovery of hundreds of thousands of realistic transit signatures injected at different ephemerides into the light curve of a single star. In this case, the FLTI code is used to independently calculate an empirical window function that can be used to verify the stand-alone, augmented TPS module's window function calculation. This verification is only practical for a small set of stars because FLTI is orders of magnitude more costly due to the large number of injections required to infer the window function. However, once verified, the augmented TPS code can readily calculate accurate window functions for all *Kepler* targets.

Figure 4 shows the results of the initial verification exercise, comparing the FLTI and augmented TPS window functions for four different stars. The initial verification identified differences as large as 50% between the FLTI window functions (black lines) and the initial augmented TPS window functions (red lines). This discrepancy was traced to the fact that the three-transit weight check was missing from the augmented TPS window function code (*i.e.*, the code only accounted for the three-transit requirement). With the addition of an extra array to retain the results of the weight checks for three-transit events, the final augmented TPS window functions for DR25 (blue lines) are

consistent with the empirical window functions (black lines) computed with FLTI to within the sampling errors of these Monte-Carlo results.

As a cautionary note, the three-transit weight check was also missing from the augmented TPS code used to produce the DR24 window functions (Burke & Seader 2016). At that time, the FLTI code was not available for verification, so this discrepancy was not identified. Consequently, the archival window functions for DR24 are overly optimistic representations of the actual SOC 9.2 pipeline performance. End users of the DR24 window functions are encouraged to consider how this discrepancy might impact their scientific results.

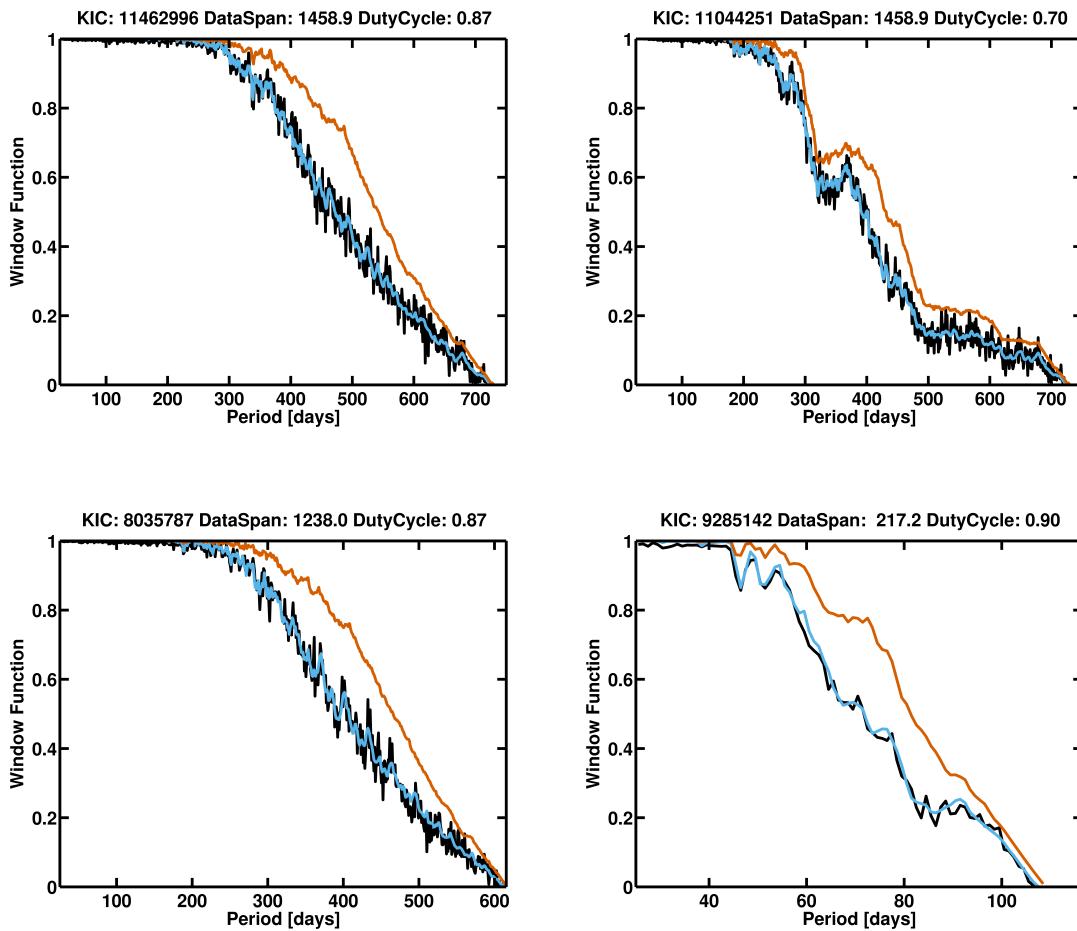


Figure 4. Example window functions of representative targets with varying amounts of observational data. The target's *Kepler* ID, data span, and duty cycle are specified along the top. The previous window function delivery (red line) for DR24 (SOC 9.2) is overly optimistic compared to the empirically derived window function (black line) computed with FLTI using a Monte-Carlo technique. The current window function delivery (DR25; SOC 9.3 – blue line) is in agreement with the empirical window function after including the three-transit weight check.

5. References

Borucki, W. J., Koch, D. G., Batalha, N., et al. 2012, *ApJ*, 745, 120

Burke, C. J., Gaudi, B. S., DePoy, D. L., & Pogge, R. W. 2006, *AJ*, 132, 210

Burke, C. J., & McCullough, P. R. 2014, *ApJ*, 792, 79

Burke, C. J., & Seader, S. E. 2016, Planet Detection Metrics: Window and One-Sigma Depth Functions (KSCI-19085-002)

Burke, C. J., & Catanzarite, J. 2017, Planet Detection Metrics: Completeness Contour Model for Data Release 25 (KSCI-19111-001)

Christiansen, J. L., Jenkins, J. M., Caldwell, D. A., et al. 2012, *PASP*, 124, 1279

Christiansen, J. L., Clarke, B. D., Burke, C. J., et al. 2016, *ApJ*, 828, 99

Gaudi, B. S. 2000, *ApJL*, 539, L59

Jenkins, J. M. 2002, *ApJ*, 575, 493

Jenkins, J. M., et al. 2017, *Kepler* Data Processing Handbook (KSCI-19081-002)

O'Donovan, F. T., Charbonneau, D., Mandushev, G., et al. 2006, *ApJ*, 651, L61

Tenenbaum, P., Jenkins, J. M., Seader, S., et al. 2014, *ApJS*, 211, 6

Thompson, S. E. 2016, Data Validation Time Series File: Description of File Format and Content (KSCI-19079-001)

Thompson, S. E., Caldwell, D. A., Jenkins, J. M., et al. 2016, *Kepler* Data Release 25 Notes for Q0-Q17 (KSCI-19065-002)

Twicken, J. D., Jenkins, J. M., Seader, S. E., et al. 2016, *AJ*, 152, 158

Appendix A: Column Definitions for Related Products

In addition to the window and one-sigma depth functions described in this document, the computation of pipeline completeness contours requires a number of other parameters (see Burke & Catanzarite 2017). Since these parameters are required for every observed star in a selected sample, they are included in the DR 25 Stellar Table at the NExSci Exoplanet Archive. Their column definitions are grouped under the heading “Occurrence Rate Columns” on the “Data Columns in the Kepler Stellar Page” at the Archive and are reproduced here for the convenience of our users.

Database Column Name	Table Label	Description
dutycycle	Duty Cycle	The fraction of data cadences within the span of observations that contain valid data and contribute toward detection of transit signals. The values for Duty Cycle ranges from 0.0 and 1.0.
dutycycle_post	Duty Cycle Post Planet Removal	The duty cycle calculated after the transit search is complete and all transit signals have been removed from the light curve. This metric is useful in identifying targets where large amounts of data were removed as a result of multiple planet search iterations.
dataspan	Data Span [days]	The time elapsed in days between the first and last cadences containing valid data.
dataspan_post	Data Span Post Planet Removal [days]	The data span calculated after the transit search is complete and all transit signals have been removed from the light curve. This metric is useful in identifying targets where large amounts of data were removed as a result of multiple planet search iterations.
mesthres01p5	Multiple Event Statistic (MES) threshold for a searched transit duration of 1.5 hours	The Multiple Event Statistic (MES) threshold reports the transit-signal significance level achieved by the transiting planet search (TPS) module. There is one entry for each of the 14 transit durations (1.5, 2.0, 2.5, 3.0, 3.5, 4.5, 5.0, 6.0, 7.5, 9.0, 10.5, 12.0, 12.5, 15.0 hr) explored by TPS in the transit search. A value of 7.1 indicates that TPS reached the nominal search threshold significance, whereas a value above 7.1 indicates that TPS ended the search prematurely at the higher specified significance threshold.
mesthres02p0	MES threshold: 2.0 hours	
mesthres02p5	MES threshold: 2.5 hours	
mesthres03p0	MES threshold: 3.0 hours	
mesthres03p5	MES threshold: 3.5 hours	
mesthres04p5	MES threshold: 4.5 hours	
mesthres05p0	MES threshold: 5.0 hours	
mesthres06p0	MES threshold: 6.0 hours	
mesthres07p5	MES threshold: 7.5	

	hours	
mesthres09p0	MES threshold: 9.0 hours	
mesthres10p5	MES threshold: 10.5 hours	
mesthres12p0	MES threshold: 12.0 hours	
mesthres12p5	MES threshold: 12.5 hours	
mesthres15p0	MES threshold: 15.0 hours	
rrmscdpp01p5	Robust RMS CDPP for a searched transit duration of 1.5 hours	<p>The robust root-mean-square (RMS) combined differential photometric precision (CDPP) is an empirical estimate of the noise in the relative flux time series observations. TPS computes a non-stationary time series of CDPP, which sets the significance level of detected transit signals.</p> <p>The 14 entries are the CDPP time series summary statistics for 14 transit durations (i.e., 1.5, 2.0, 2.5, 3.0, 3.5, 4.5, 5.0, 6.0, 7.5, 9.0, 10.5, 12.0, 12.5, 15.0 hr) searched by TPS. These summary statistics are calculated in a robust fashion by excluding cadences de-weighted during the transit search and employing median and mean absolute deviation algorithms, rather than the less robust arithmetic mean and standard deviation. Specifically, the standard definition of the root mean square deviation of a data set X is $X_{rms}^2 = X_{mean}^2 + X_{std}^2$, where X_{mean} is the arithmetic mean of X and X_{std} is the standard deviation of X. For the robust version, the arithmetic mean is replaced with the median and the standard deviation is replaced with the mean absolute deviation (mad) scaled to approximate the standard deviation, so that $(X_{robust_rms})^2 = (X_{median})^2 + (1.4826 * X_{mad})^2$.</p>
rrmscdpp02p0	Robust RMS CDPP: 2.0 hours	
rrmscdpp02p5	Robust RMS CDPP: 2.5 hours	
rrmscdpp03p0	Robust RMS CDPP: 3.0 hours	
rrmscdpp03p5	Robust RMS CDPP: 3.5 hours	
rrmscdpp04p5	Robust RMS CDPP: 4.5 hours	
rrmscdpp05p0	Robust RMS CDPP: 5.0 hours	
rrmscdpp06p0	Robust RMS CDPP: 6.0 hours	
rrmscdpp07p5	Robust RMS CDPP: 7.5 hours	
rrmscdpp09p0	Robust RMS CDPP: 9.0 hours	
rrmscdpp10p5	Robust RMS CDPP: 10.5 hours	
rrmscdpp12p0	Robust RMS CDPP: 12.0 hours	
rrmscdpp12p5	Robust RMS CDPP: 12.5 hours	
rrmscdpp15p0	Robust RMS CDPP: 15.0 hours	
cdppslplong	rmsCDPP Slope for Long Transit Durations	The rmsCDPP slope for long transit durations is a linear fit to the log10(transit duration) versus log10(rmsCDPP) values for the six longest

		<p>transit durations (e.g. 7.5, 9.0, 10.5, 12.0, 12.5, 15.0 hr).</p> <p>Note: For white Gaussian noise, the rmsCDPP is expected to decrease with increasing transit duration because the noise decreases as the square-root of the number of in-transit cadences. However, targets with astrophysical noise or non-uniform power spectral densities can have rmsCDPP values that actually increase with increasing transit duration. Hence, this rmsCDPP slope serves as an indicator of non-white Gaussian noise behavior on the longest timescales searched. Monte-Carlo transit injection and recovery tests of the Kepler pipeline reveal that targets with large values of this rmsCDPP slope have suppressed transit recovery relative to targets with nominal rmsCDPP slopes.</p>
cdppslpshrt	rmsCDPP Slope for Short Transit Duration	See the definition of rmsCDPP slope for long transit duration. This metric is determined by fitting the slope of the second to sixth shortest transit durations (e.g. 2.0, 2.5, 3.0, 3.5, 4.5 hr) in order to characterize the noise on short timescales. This fit differs from the long transit duration slope in that it is performed in a semi-log space (e.g. $\log_{10}(\text{transit duration})$ versus rmsCDPP).
timeout01p5	Timeout Indicator for a searched transit duration of 1.5 hours	<p>The timeout indicators report whether the TPS planet search completed normally for the 14 transit durations. The searches in the TPS proceed from the longest transit duration to the shortest. The searches at individual durations end when their allocated share of resources is consumed and the entire search process ends when an overall timeout is reached.</p> <p>Timeout indicator values:</p> <ul style="list-style-type: none"> • 0 (zero) = the search ended normally after reaching its nominal significance threshold of 7.1; • 1 = the search ended prematurely, before the 7.1 significance threshold was achieved; • 2 = no transit search was attempted.
timeout02p0	Timeout Indicator: 2.0 hours	
timeout02p5	Timeout Indicator: 2.5 hours	
timeout03p0	Timeout Indicator: 3.0 hours	
timeout03p5	Timeout Indicator: 3.5 hours	
timeout04p5	Timeout Indicator: 4.5 hours	
timeout05p0	Timeout Indicator: 5.0 hours	
timeout06p0	Timeout Indicator: 6.0 hours	
timeout07p5	Timeout Indicator: 7.5 hours	
timeout09p0	Timeout Indicator: 9.0 hours	
timeout10p5	Timeout Indicator: 10.5 hours	
timeout12p0	Timeout Indicator: 12.0 hours	
timeout12p5	Timeout Indicator: 12.5 hours	
timeout15p0	Timeout Indicator:	

	15.0 hours	
timeoutsumry	Timeout Indicator Summary	This is a binary flag where a value of 1 indicates that all 14 transit durations were searched down to the 7.1 significance threshold with no timeout issues, and a value of 0 (zero) indicates that at least one of the transit durations did not achieve a search down to the 7.1 significance threshold.